

CPS216: Data-Intensive Computing Systems

Query Processing (Overview)

Shivnath Babu

Query Processing

Declarative **SQL** Query → Query Plan

NOTE: You will not be tested on how well you know SQL. Understanding the SQL introduced in class will be sufficient (a primer follows). SQL is described in Chapter 6, GMUW.

Focus: Relational System (i.e., data is organized as tables, or relations)

SQL Primer

We will focus on SPJ, or Select-Project-Join Queries

Select <attribute list>

From <relation list>

Where <condition list>

Example Filter Query over R(A,B,C):

Select B

From R

Where R.A = "c" \wedge R.C > 10

SQL Primer (contd.)

We will focus on SPJ, or Select-Project-Join-Queries

Select <attribute list>

From <relation list>

Where <condition list>

Example Join Query over R(A,B,C) and S(C,D,E):

Select B, D

From R, S

Where $R.A = \text{"c"} \wedge S.E = 2 \wedge R.C = S.C$

R	A	B	C	S	C	D	E
a	1	10	10	10	x	2	
b	1	20	20	20	y	2	
c	2	10	30	30	z	2	
d	2	35	40	40	x	1	
e	3	45	50	50	y	3	

Select B,D

From R,S

Where $R.A = "c" \wedge$

$S.E = 2 \wedge R.C = S.C$

Answer

B	D
2	x

- How do we execute this query?

Select B,D

From R,S

Where $R.A = \text{"c"} \wedge S.E = 2 \wedge$

$R.C = S.C$



One idea

- Do Cartesian product
- Select tuples
- Do projection

R X S

R.A	R.B	R.C	S.C	S.D	S.E
a	1	10	10	x	2
a	1	10	20	y	2
.					
.					
c	2	10	10	x	2
.					
.					

Select B,D

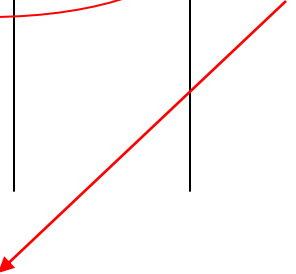
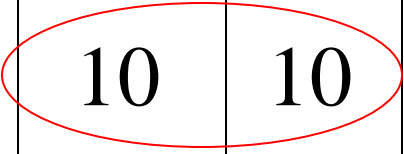
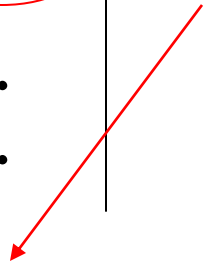
From R,S

Where R.A = "c"

\wedge S.E = 2 \wedge

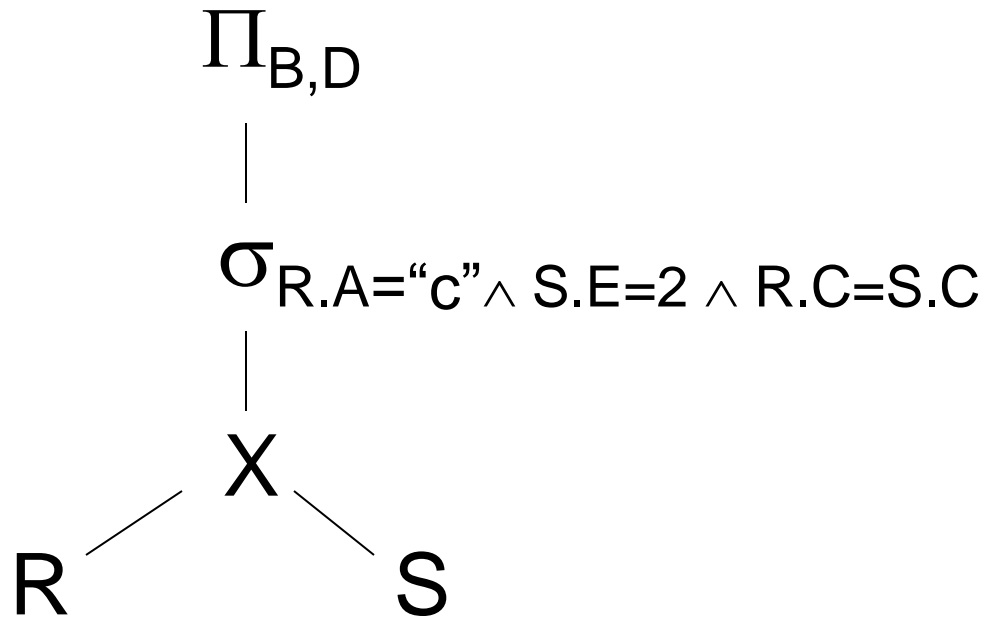
R.C=S.C

Bingo! →
Got one...



Relational Algebra - can be used to describe plans

Ex: Plan I



Relational Algebra Primer (Chapter 5, GMUW)

Select: $\sigma_{R.A="c" \wedge R.C=10}$

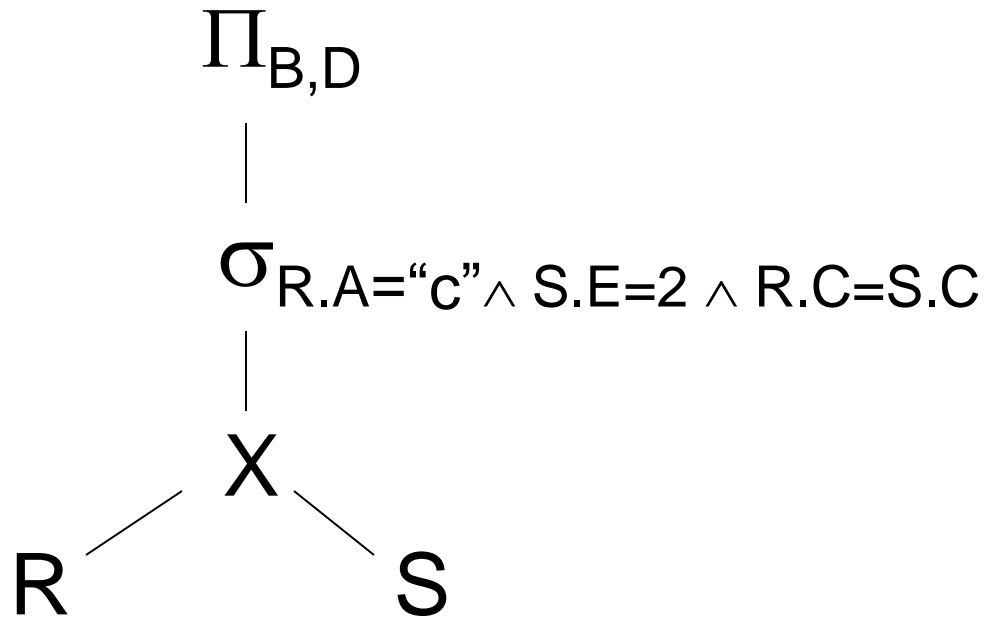
Project: $\Pi_{B,D}$

Cartesian Product: $R \times S$

Natural Join: $R \bowtie S$

Relational Algebra - can be used to describe plans

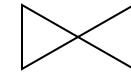
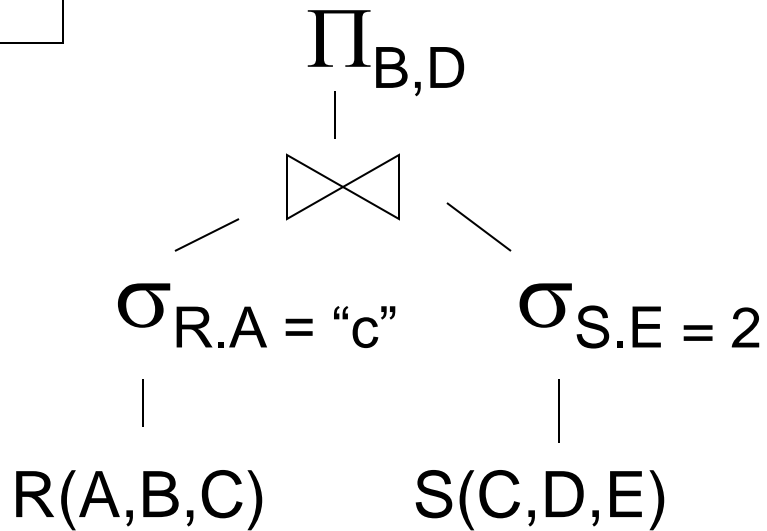
Ex: Plan 1



OR: $\Pi_{B,D} [\sigma_{R.A="c" \wedge S.E=2 \wedge R.C=S.C} (RXS)]$

Another idea:

Plan II



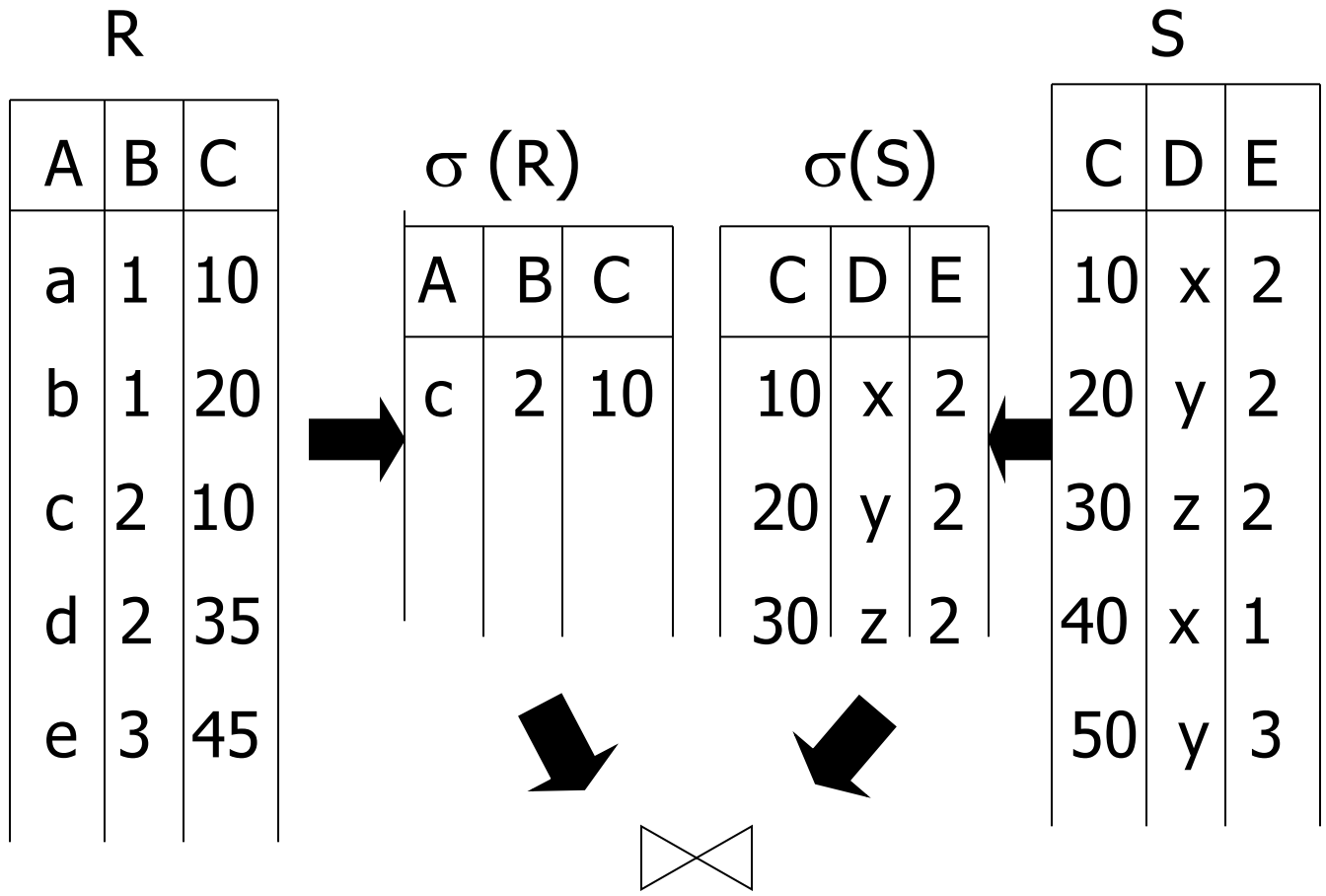
natural join

Select B,D

From R,S

Where $R.A = \text{"c"} \wedge$

$S.E = 2 \wedge R.C = S.C$



Select B,D

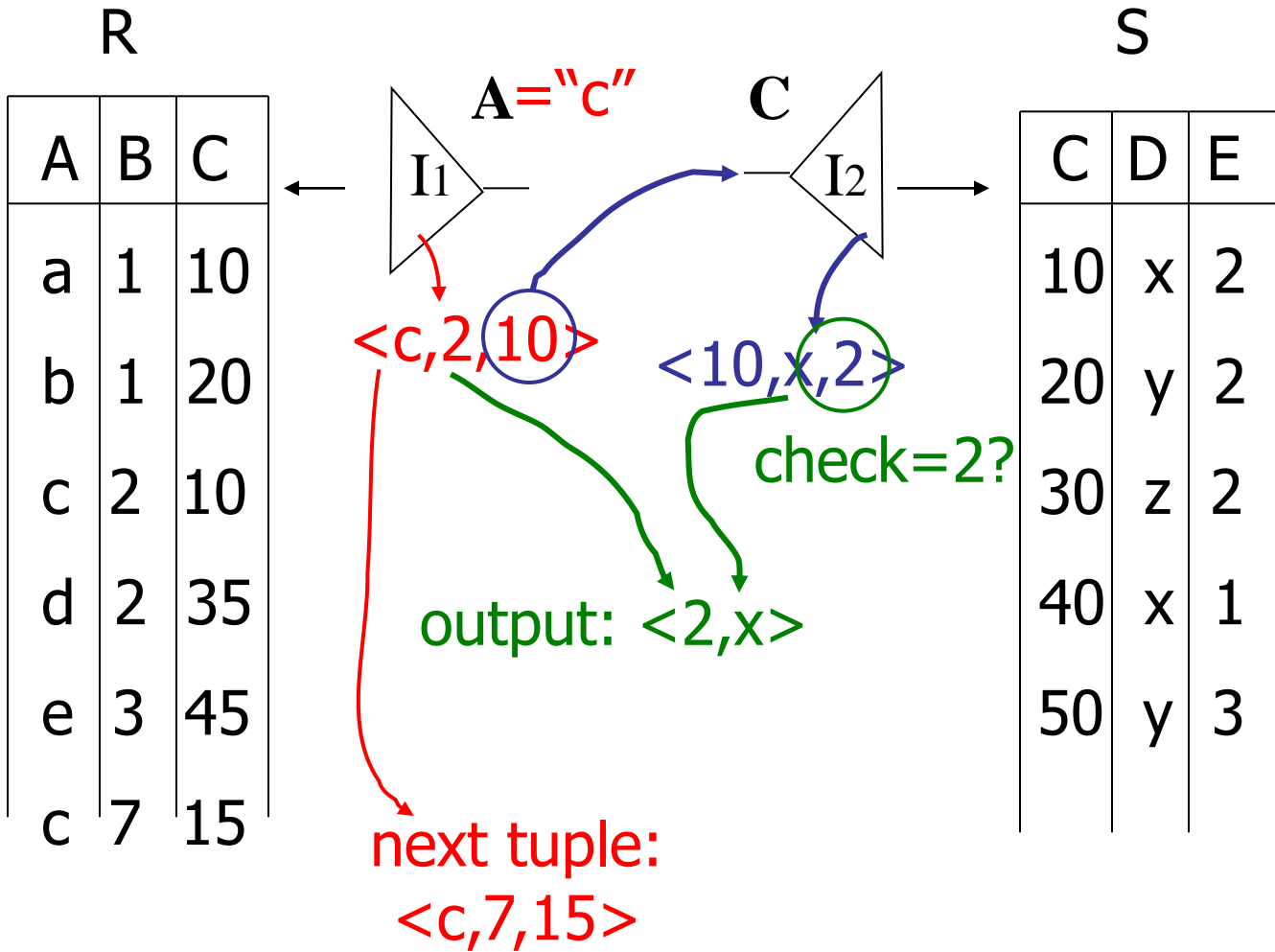
From R,S

Where R.A = "c" \wedge
 S.E = 2 \wedge R.C=S.C

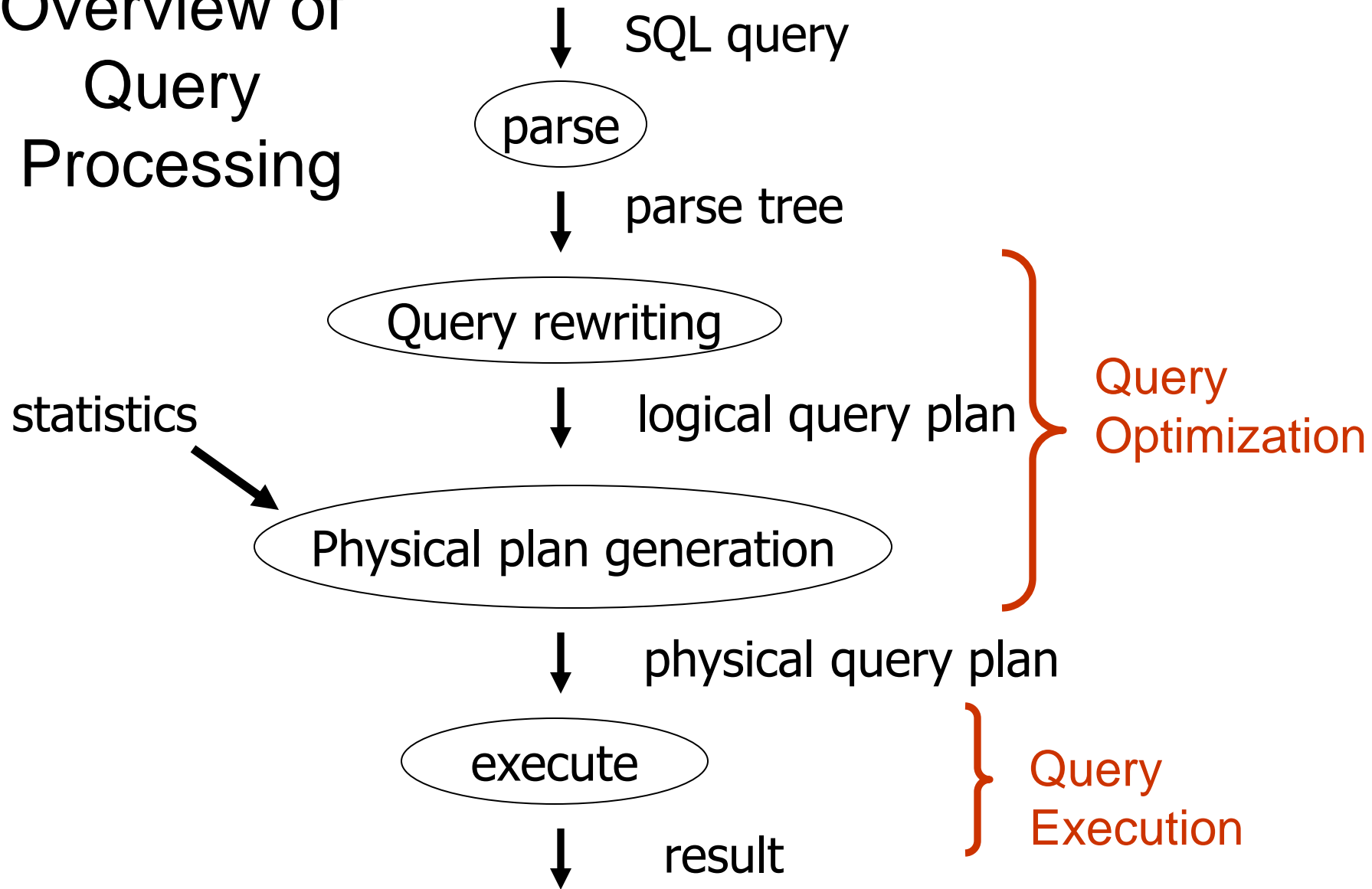
Plan III

Use R.A and S.C **Indexes**

- (1) Use R.A index to select R tuples with R.A = "c"
- (2) For each R.C value found, use S.C index to find matching tuples
- (3) Eliminate S tuples S.E \neq 2
- (4) Join matching R,S tuples, project B,D attributes, and place in result



Overview of Query Processing



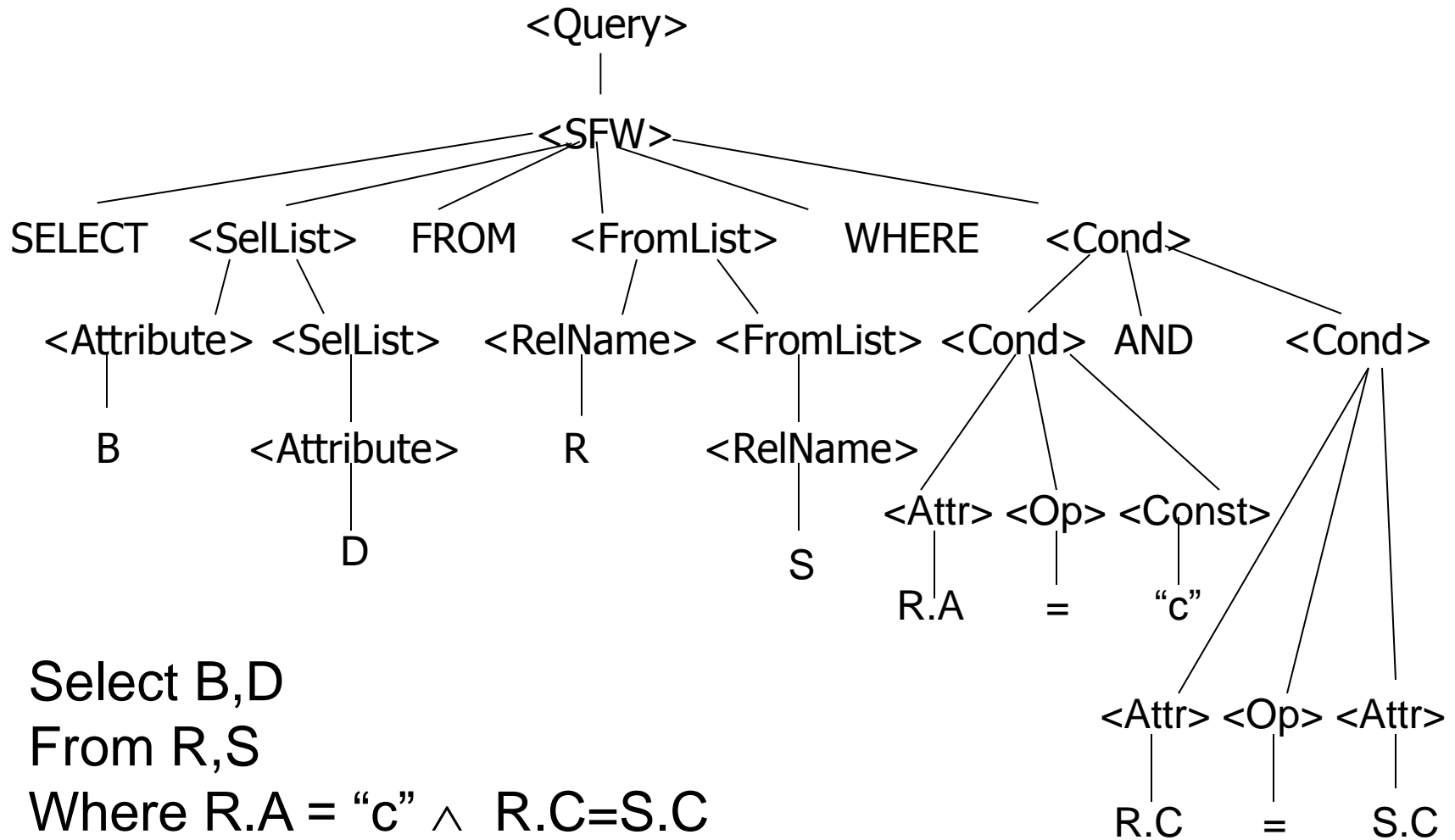
Example Query

Select B,D

From R,S

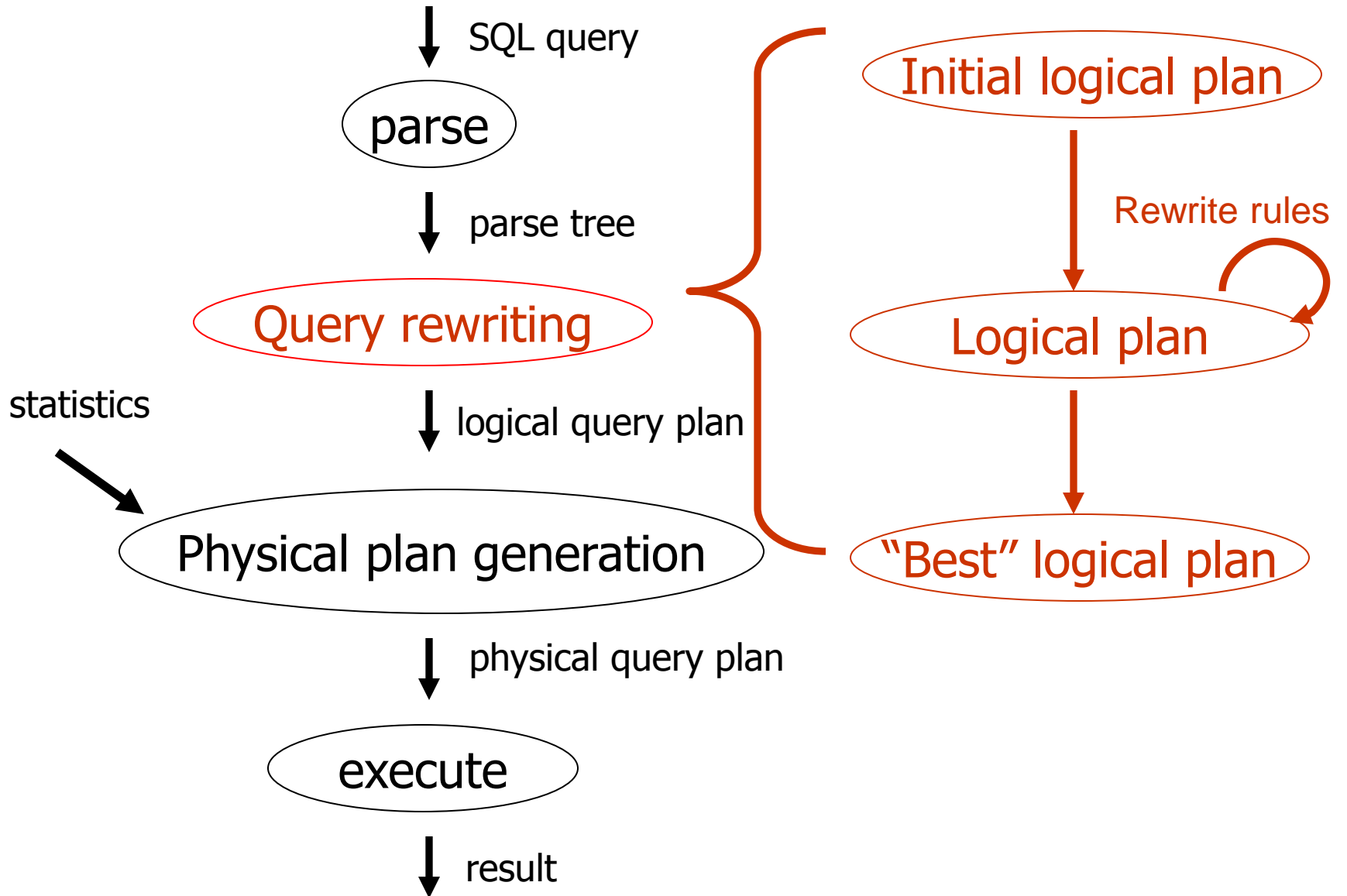
Where $R.A = \text{"c"} \wedge R.C = S.C$

Example: Parse Tree



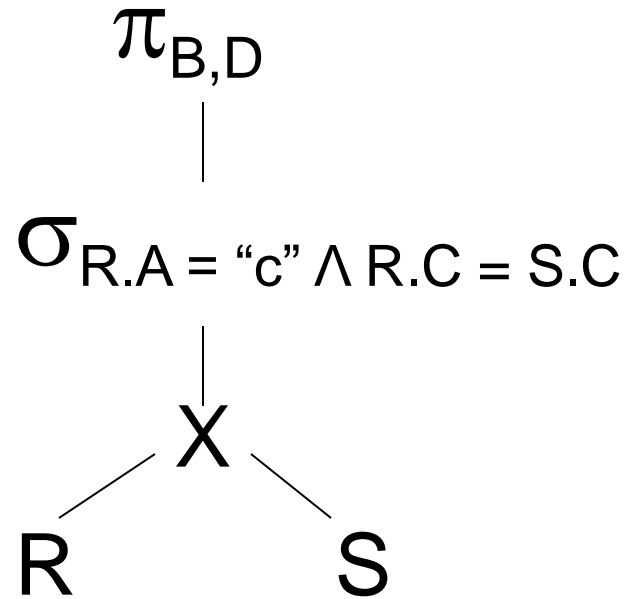
Along with Parsing ...

- Semantic checks
 - Do the projected attributes exist in the relations in the From clause?
 - Ambiguous attributes?
 - Type checking, ex: $R.A > 17.5$
- Expand **views**



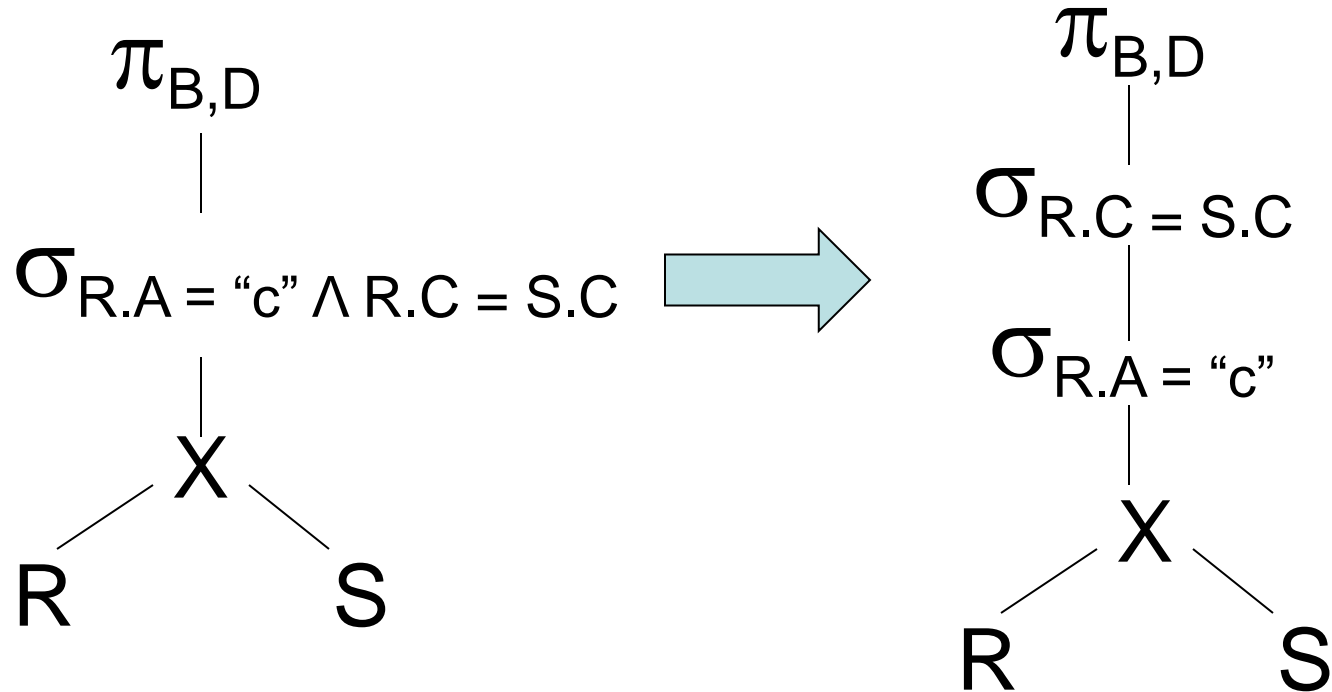
Initial Logical Plan

Select B,D
From R,S
Where R.A = "c" \wedge
R.C=S.C



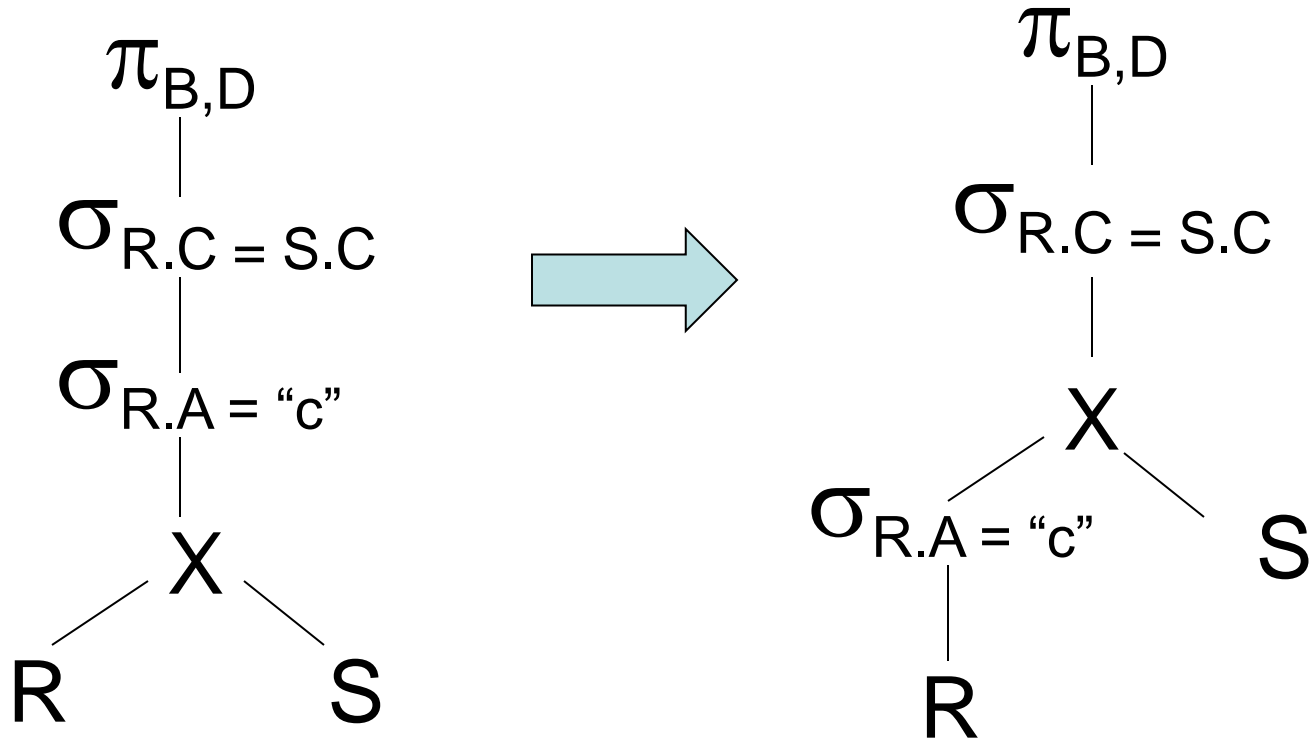
Relational Algebra: $\Pi_{B,D} [\sigma_{R.A = \text{"c"} \wedge R.C = S.C} (RXS)]$

Apply Rewrite Rule (1)



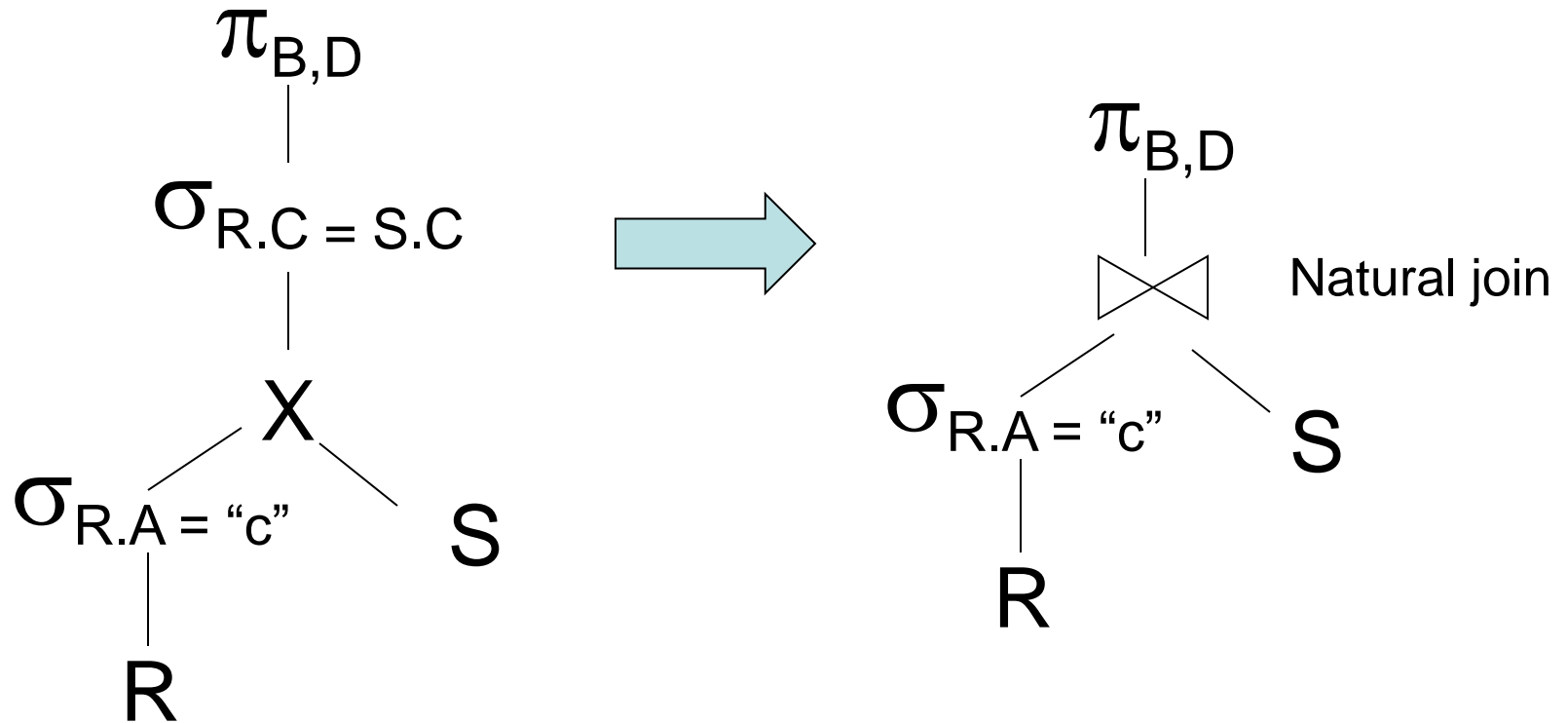
$$\Pi_{B,D} [\sigma_{R.C=S.C} [\sigma_{R.A=\text{"c"}}(R X S)]]$$

Apply Rewrite Rule (2)



$\Pi_{B,D} [\sigma_{R.C=S.C} [\sigma_{R.A=\text{"c"}}(R)] X S]$

Apply Rewrite Rule (3)



$$\Pi_{B,D} [[\sigma_{R.A='c'}(R)] \bowtie S]$$

